Simulating Quasispecies Composition

Gregori, J and Guerrero, M

09/05/2018

Contents

1 Introduction

A viral quasispecies is understood as a collection of closely related viral genomes produced by viruses with low replication fidelity. RNA viruses show a high replication error rate due to a lack of proofreading mechanisms. It is estimated that for viruses with typically high replicative loads, every possible point mutation and many double mutations are generated with each viral replication cycle, and these may be present within the population at any time. Given this inherent dynamic, we may be interested in comparing the viral diversity indices between sequential samples from a single patient or between samples from groups of patients. These comparisons can provide information on the patient's clinical progression or the appropriateness of a given treatment.

QSUtils is a package intended for use with quasispecies amplicon data obtained by NGS, but it could also be useful for analyzing 16S/18S ribosomal-based metagenomics or tumor genetic diversity by amplicons.

In this tutorial, we illustrate how the functions provided in the package can be used to simulate quasispecies data. This implies simulation of closely related genomes, (eventually with segregating subpopulations at higher genetic distances) and their abundances.

In particular, we can differentiate between acute and chronic infection profiles by the quasispecies composition. An acute infection is expected to show a prominent genome that is highly abundant, together with a set of low-abundance genomes. On the other hand, besides the implicit dynamics, a chronic infection is expected to show a number of relatively abundant genomes together with a myriad of derived genomes at low and very low abundances.

In viral terms, the fitness of a genome is a measure of its replicative performance. High-fitness haplotypes show high abundances after a transient state, during which they overcome other genomes in the quasispecies. During infection of a human host, the quasispecies typically shows variations in the fitness of each genome caused by changes in the bioenvironment. Because of this dynamic, we may observe the profile of a typical acute infection also in chronic patients, at least at the level of magnification provided by current NGS technology.

A few functions in the package were designed to simulate quasispecies composition, with the aim of studying the statistical properties of the diversity indices (Gregori et al. 2016) (Gregori et al. 2014).

2 Install package

```
library(devtools)
install_git("https://github.com/VHIRHepatiques/QSutils")

library(QSutils)
```

3 Abundance

Two different types of information define the quasispecies composition: the genomes present and their current frequency (abundance) in the viral population. The package provides three functions to simulate abundance with various decreasing profiles.

3.1 Powers of a fraction

$fn.ab.1$ computes consecutive fractions, given the frequency of the most abundant haplotype, according to:

$$h \; r^{(i-1)}, \quad r < 1, \quad i = 1..n$$

Using $table$ on the $fn.ab.1$ result, we obtain the number of haplotypes for each frequency:

```
table(fn.ab.1(25))
##
##     1     2     4     9    19    39    78   156   312   625  1250  2500
##    12     1     1     1     1     1     1     1     1     1     1     1
##  5000 10000
##     1     1
```

```
par(mfrow=c(1,2))
plot(fn.ab.1(25),type="h")
plot(fn.ab.1(25,r=0.7),type="h")
```

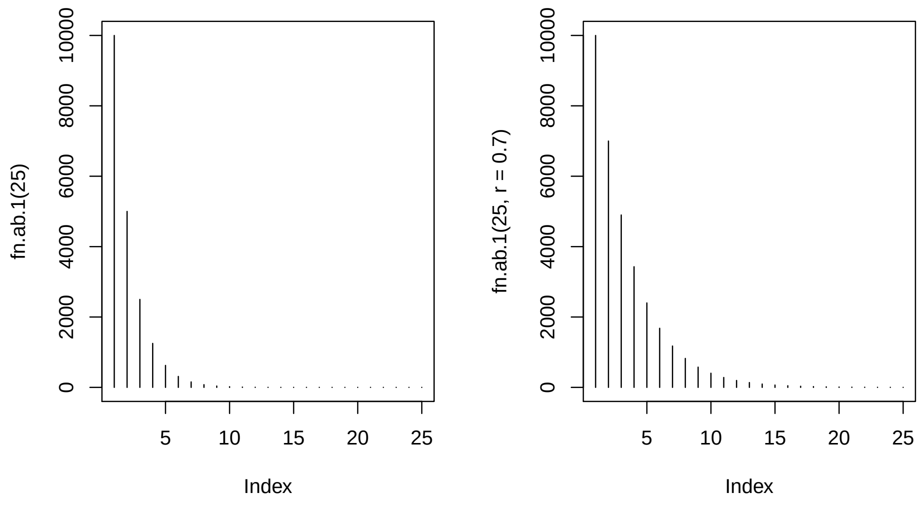


Figure 1: Profile of abundances simulated with ‘fn.ab.1‘

The default value for r is 0.5. Higher r values moderate the decrease of abundances. By default, the frequency of the most abundant haplotype, h, is 10000.

3.2 Power of consecutive fractions

$fn.ab.2$ computes the power of consecutive fractions, according to:

$$h\,\frac{1}{i^r},\quad r>0,\quad i=1..n$$

Default values are 3 for r and 10,000 for h. The higher the r, the more pronounced the decrease in frequencies.

```
table(fn.ab.2(25))
##
##     1     2     3     4     5     7    10    13    19    29    46    80
##     8     3     1     1     1     1     1     1     1     1     1     1
##   156   370  1250 10000
##     1     1     1     1
```

```
table(fn.ab.2(25,r=2))
##
##    16    17    18    20    22    25    27    30    34    39    44    51
##     1     1     1     1     1     1     1     1     1     1     1     1
##    59    69    82   100   123   156   204   277   400   625  1111  2500
##     1     1     1     1     1     1     1     1     1     1     1     1
## 10000
##     1
```

```
par(mfrow=c(1,2))
plot(fn.ab.2(25),type="h")
plot(fn.ab.2(25,r=2),type="h")
```

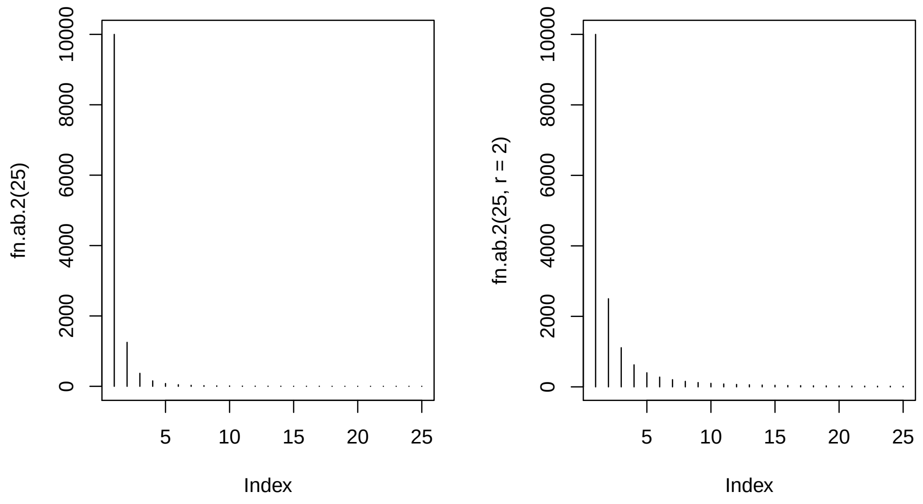


Figure 2: **Abundances simulated with ‘fn.ab.2‘**

3.3 Decreasing fractional powers

$fn.ab.3$ computes decreasing roots of the maximum frequency, h, according to:

$$h^{1/i}, \quad i = 1..n$$

```
table(fn.ab.3(25))
##
##     1     2     3     4     6    10    21   100 10000
##    12     5     2     1     1     1     1     1     1
```

```
par(mfrow=c(1,2))
plot(fn.ab.3(25),type="h")
```

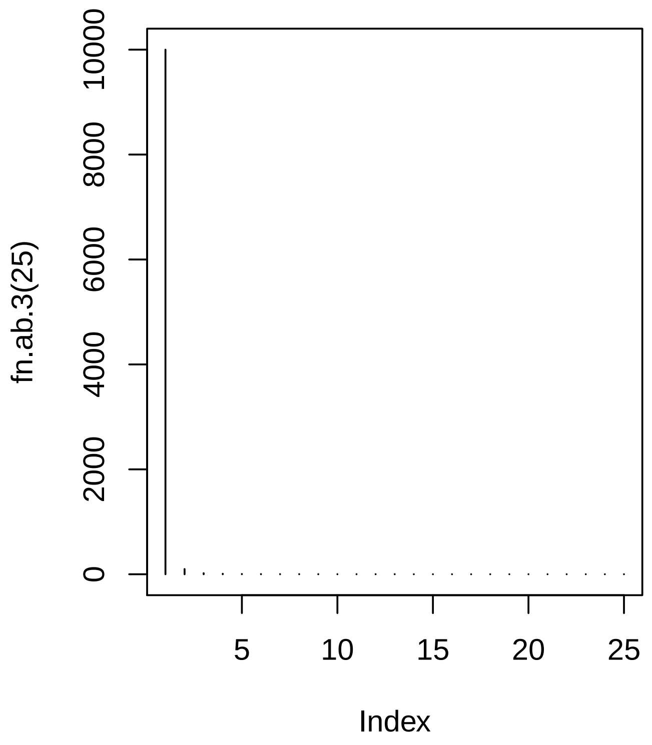


Figure 3: Abundances simulated with ‘fn.ab.3‘

The figure and the previous table both show that this function is the one that generates the greatest distance between the dominant haplotype and the others.

To compare the profiles of the three functions, this figure plots the outputs of the functions with default parameters.

```
par(mfrow=c(1,3))
plot(fn.ab.1(25),type="h",main="fn.ab.1")
plot(fn.ab.2(25),type="h",main="fn.ab.2")
plot(fn.ab.3(25),type="h",main="fn.ab.3")
```

A linear combination of results of the three functions provides greater flexibility.

```
ab <- 0.25*fn.ab.1(25)+0.75*fn.ab.2(25)
table(ab)
## ab
```

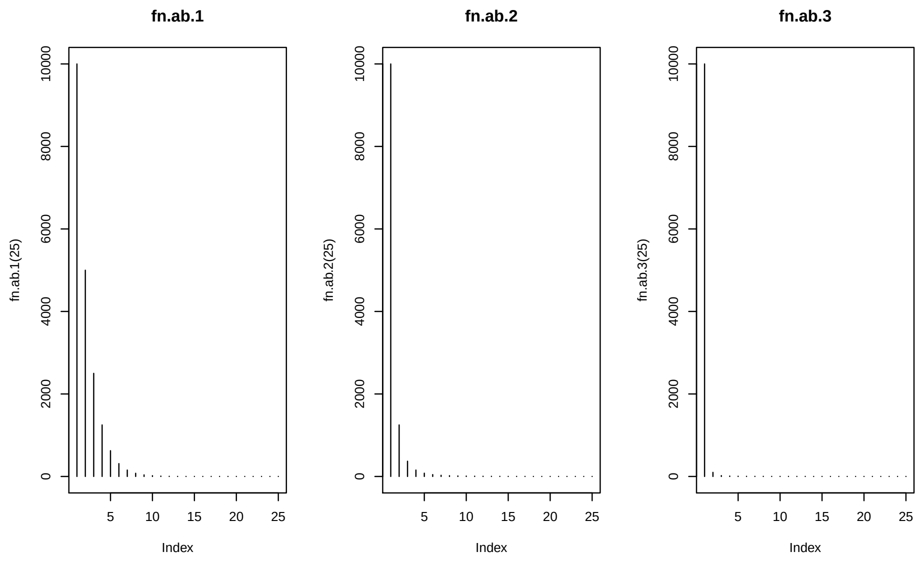


Figure 4: Comparison of the data simulation functions

```
##      1   1.75    2.5    3.5   4.75    7.5  12.25   19.5  33.75  60.75  112.5
##      8      3      1      1      1      1      1      1      1      1      1
## 216.25  429.5  902.5 2187.5  10000
##      1      1      1      1      1
```

```
plot(ab,type="h",main="Linear combination of results")
```

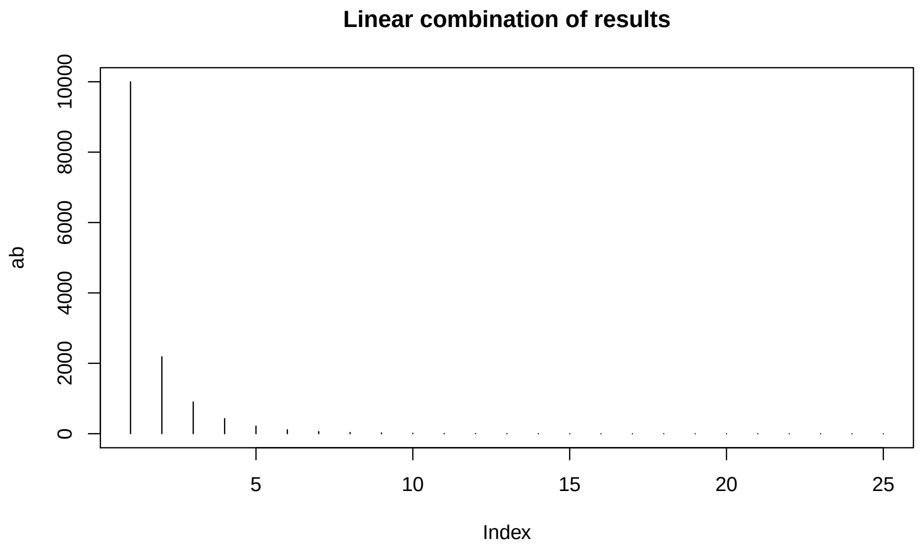


Figure 5: Abundances simulated with a linear combination of the functions

```
ab <- 0.7*fn.ab.1(25)+0.3*fn.ab.3(25)
table(ab)
## ab
##      1      2    3.4    6.9   13.9   27.9   55.5  110.1  219.6  439.3    878
##      12       1       1       1       1       1       1       1       1       1       1
## 1756.3    3530   10000
##       1       1       1
```

```
plot(ab,type="h",main="Linear combination of results")
```

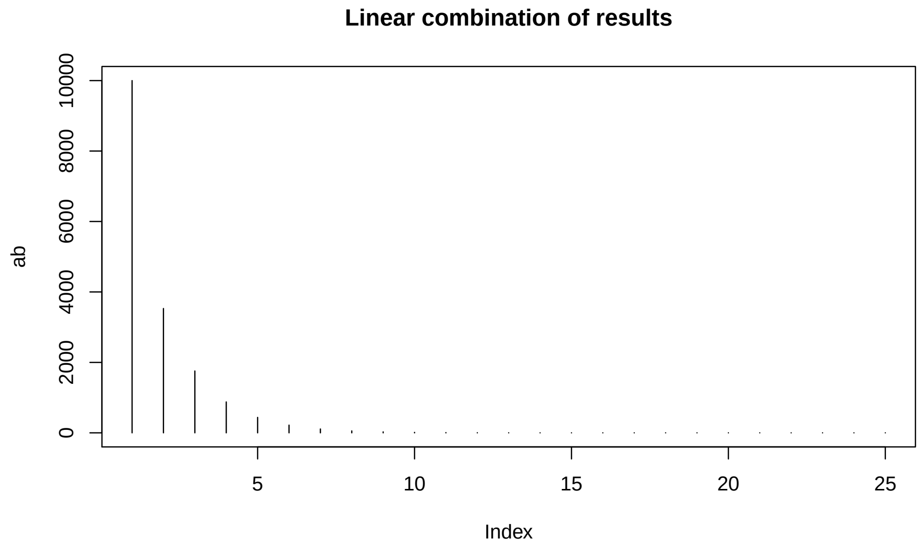


Figure 6: **Abundances simulated with a linear combination of the functions**

3.4 Geometric sequence

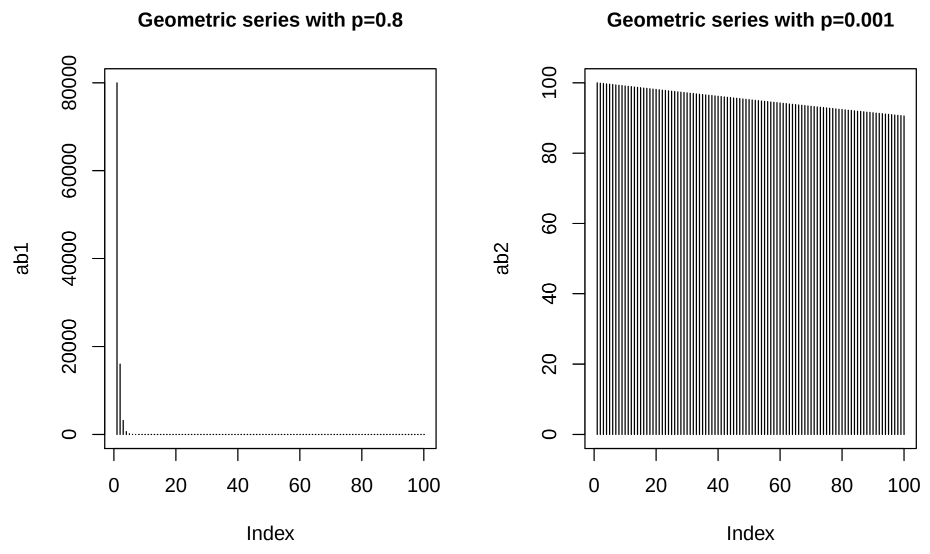
Appropriate for the typical load of rare haplotypes observed in our experiments with the HCV quasispecies before the abundance filter. That is, the large number of very low fitness or defective haplotypes are best simulated by a geometric sequence with low values for the parameter. The geometric sequence is expressed as:

$$p\ (1-p)^{k-1},\quad k=1..n,\quad 0<p<1$$

and is implemented in the function $geom.series$, taking two arguments: n, the number of frequencies to compute and p, the parameter of the geometric function.

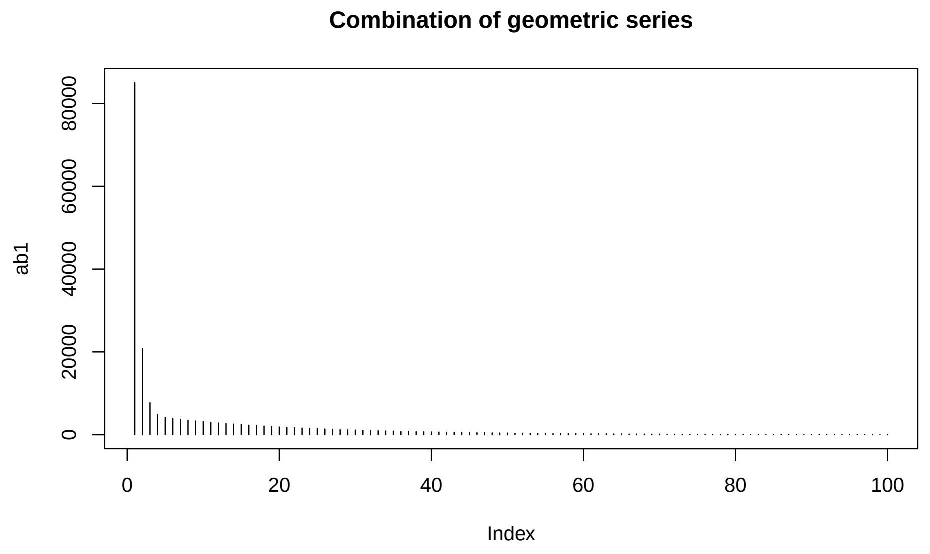
This function is useful to simulate a broad spectrum of frequency profiles, from quasispecies with very prevalent haplotypes to the above-mentioned long queues of very low abundances, as illustrated in the next figure.

```
par(mfrow=c(1,2))
ab1 <- 1e5 * geom.series(100,0.8)
plot(ab1,type="h",main="Geometric series with p=0.8",cex.main=1)
ab2 <- 1e5 * geom.series(100,0.001)
plot(ab2,type="h",main="Geometric series with p=0.001",ylim=c(0,max(ab2)),
    cex.main=1)
```



Linear combinations of geometric sequences with parameters of differing magnitudes help to obtain typical quasispecies profiles:

```
ab1 <- 1e5 * (geom.series(100,0.8)+geom.series(100,0.05))
plot(ab1,type="h",main="Combination of geometric series")
```



The functions $fn.ab.1$, $fn.ab.2$, and $fn.ab.3$ are flexible enough to obtain typical quasispecies profiles after filtering out all haplotypes below an abundance threshold, considered the technical noise level. These functions, combined with geometric series having low to very low parameter values, provide profiles close to those observed empirically.

4 Random genomes and variant haplotypes

In addition to frequencies, we need to simulate the quasispecies genomes. The first task for this purpose is to generate the dominant haplotype by $GetRandomSeq$. The only parameter for this function is the genome length. The output is a fully random sequence of nucleotides, returned as a character string.

```
set.seed(23)
m1 <- GetRandomSeq(50)
m1
## [1] "GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT"
```

Variant genomes of this haplotype can be generated by $GenerateVars$. This function takes four parameters, seq the dominant haplotype, $nhpl$ the number of variants to generate, $max.muts$ the maximum number of mutations in a genome, and $p.muts$ the probability for each number of mutations from 1 to $max.muts$. It returns a vector of character strings with the variant genomes.

```
v1 <- GenerateVars(m1,20,2,c(10,1))
DottedAlignment(c(m1,v1))
##  [1] "GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT"
##  [2] "............................T....................."
##  [3] ".............................................G...."
##  [4] "....A............................................."
##  [5] "........................................A........."
##  [6] ".............................C...................."
##  [7] "..........................................A......."
##  [8] ".............................................C...."
##  [9] "....C....................................C........"
## [10] ".....A............................................"
## [11] "..A..............................................."
## [12] "..........................A......................."
## [13] "........................A........................."
## [14] "..................................G..............."
## [15] "....................C............................."
## [16] "............................................G....."
## [17] "....................C............................."
## [18] ".T................................................"
## [19] ".....................................C...........A"
## [20] ".........G........................................"
## [21] "..............G..........C........................"
```

4.1 Generate a quasispecies of acute infection

With these functions we can simulate a quasispecies with a profile of acute infection; that is, characterized by a dominant haplotype which is fairly abundant, together with a number of haplotypes at low abundances.

```
set.seed(23)
n.genomes <- 25
```

```
m1 <- GetRandomSeq(50)
v1 <- GenerateVars(m1,n.genomes-1,2,c(10,1))
w1 <- fn.ab.2(n.genomes)
data.frame(Hpl=DottedAlignment(c(m1,v1)),Freq=w1)
##                                                    Hpl  Freq
## 1  GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT 10000
## 2  ....A.............................................  1250
## 3  ........................................A.........   370
## 4  .............................C....................   156
## 5  ..........................................A.......    80
## 6  .............................................C....    46
## 7  .........................................C........    29
## 8  ........C.........................................    19
## 9  .....A......A.....................................    13
## 10 ..........................A.......................    10
## 11 ........................A.........................     7
## 12 ..................................G...............     5
## 13 ....................C.............................     4
## 14 ............................................G.....     3
## 15 ....................C.............................     2
## 16 .T................................................     2
## 17 .....................................C............     2
## 18 ............................................A.....     1
## 19 .........A.................G......................     1
## 20 ..............................T...................     1
## 21 .................A.......................T........     1
## 22 .................T................................     1
## 23 ........................A.........................     1
## 24 .................T................................     1
## 25 .........................G........................     1
```

The quasispecies composition can be visualized using a bar plot depicting the haplotype frequencies, with haplotypes sorted by increasing number of mutations with respect to the dominant haplotype, and within the number of mutations, by decreasing order of abundance:

```
qs <- DNAStringSet(c(m1,v1))
lst <- SortByMutations(qs,w1)
qs <- lst$bseqs

cnm <- cumsum(table(lst$nm))+1
nm.pos <- as.vector(cnm)[-length(cnm)]
names(nm.pos) <- names(cnm[-1])

bp <- barplot(lst$nr,col="lavender")
axis(1,at=bp[nm.pos],labels=names(nm.pos),cex.axis=0.7)
```

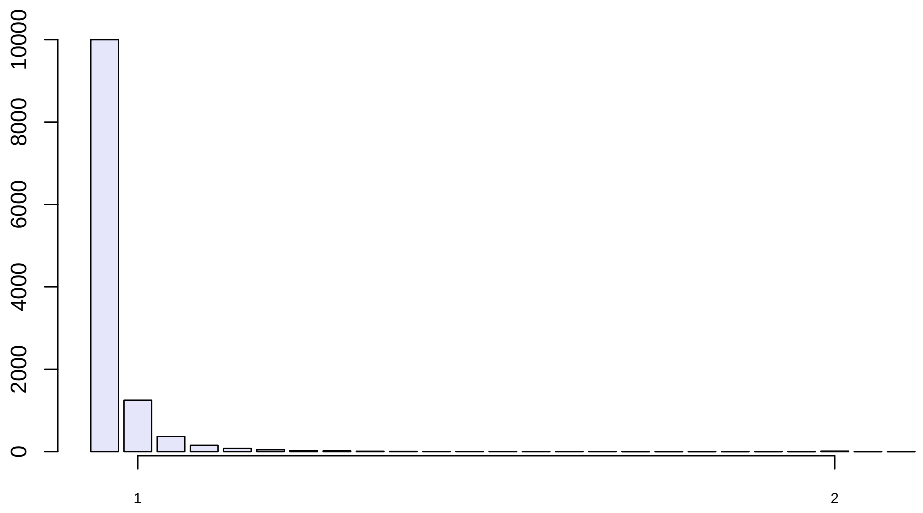


Figure 7: Simulated abundances in an acute infection

4.2 Generate a quasispecies of chronic infection

In contrast to acute infection, chronic infection develops more slowly; hence, a larger number of mutations are generated with regard to the dominant haplotype. Furthermore, the mutated haplotypes may be more abundant in chronic than in acute infections. In this case, we would use $GenerateVars$ with a higher value of $max.muts$ and higher probabilities for mutants at any level.

```
set.seed(23)
n.genomes <- 40
m1 <- GetRandomSeq(50)
v1 <- GenerateVars(m1,n.genomes-1,6,c(10,3,1,0.5,2,0.5))
w1 <- fn.ab.2(n.genomes,r=1.5)
data.frame(Hpl=DottedAlignment(c(m1,v1)),Freq=w1)
##                                                   Hpl  Freq
## 1  GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT 10000
## 2  .............A....................................  3535
## 3  .........C.....T.......A..AG......................  1924
## 4  ........................G.........................  1250
## 5  .C.....C...........................GA.........C...   894
## 6  ..........................T..G...T................   680
## 7  ...............................G..................   539
## 8  .........................................G........   441
## 9  ..............AG.C......A........A.........C......   370
## 10 ....C.............................................   316
## 11 .....G......G..............................T......   274
## 12 ...........................G......................   240
## 13 .T................................................   213
## 14 ................A.................................   190
## 15 .......C..........................................   172
## 16 ................................T.................   156
## 17 ..........................................A.......   142
## 18 ..................................A...............   130
## 19 ...............T...T...C.........T..G...C.........   120
## 20 ....................C.............................   111
## 21 .............GC..............A....A...............   103
## 22 .....................................C............    96
## 23 .................................GG........C.G...A    90
## 24 .........C......T...........................C.....    85
## 25 ................................................T.    80
## 26 .................................G................    75
## 27 ............................T.....................    71
## 28 .G......................T...T.....G.............A.    67
## 29 ...T..............................................    64
## 30 A..............T..................................    60
## 31 ......................C............C..........T...    57
## 32 ...........T.A..............G.C.....T.............    55
## 33 ......................................G...........    52
## 34 ....C...A............T............................    50
## 35 .................C.A...A.CA.......................    48
## 36 ...........A.............G................TA....T.    46
## 37 ............................................C.....    44
## 38 .............................................A....    42
## 39 ............................C........C....T.......    41
## 40 ..T...............................................    39
```

Again, we can visualize the quasispecies composition using a bar plot.

```
qs <- DNAStringSet(c(m1,v1))
lst <- SortByMutations(qs,w1)
qs <- lst$bseqs
cnm <- cumsum(table(lst$nm))+1
nm.pos <- as.vector(cnm)[-length(cnm)]
names(nm.pos) <- names(cnm[-1])
bp <- barplot(lst$nr,col="lavender")
axis(1,at=bp[nm.pos],labels=names(nm.pos),cex.axis=0.7)
```

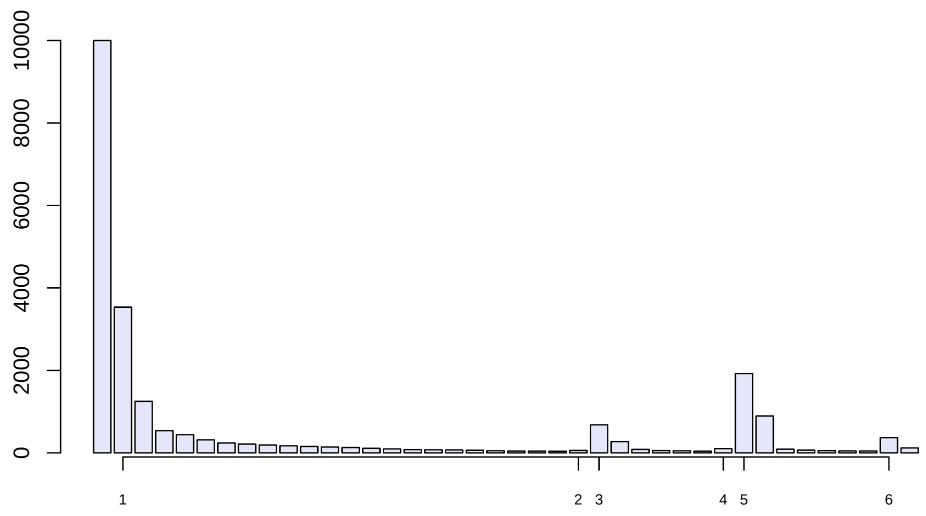


Figure 8: Simulated abundances in a chronic infection

4.3 Diverging populations

Along the quasispecies dynamics we may see emergence of a segregating subpopulation with improved fitness due to the combination of mutations, rather than a single one. In this instance, the $Diverge$ function helps by producing variants with a common pattern of mutations.

```
set.seed(23)
m1 <- GetRandomSeq(50)
p2 <- Diverge(3:5,m1)
DottedAlignment(c(m1,p2))
## [1] "GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT"
## [2] "..............C.........A...............A........."
## [3] "..............C.........A.............G.A........."
## [4] "..............C.........A.............G.AT........"
```

Variants of these sequences can be produced in the usual way by $GenerateVars$.

```
v1 <- GenerateVars(m1,20,3,c(10,4,0.2))
wv1 <- fn.ab.2(length(v1),h=1000,r=1.5)
wp2 <- c(600,1000,400)
v2 <- GenerateVars(p2[2],20,3,c(10,1,0.1))
wv2 <- fn.ab.2(length(v2),r=2,h=wp2[2]*3)

qs <-DNAStringSet(c(m1,v1,p2,v2))
w <- round(c(10000,wv1,wp2,wv2))

lst <- SortByMutations(qs,w)
qs <- lst$bseqs
data.frame(Hpl=DottedAlignment(qs),nr=lst$nr)
##                                                            Hpl    nr
## Hpl_0_0001 GACGTCTTTTTGCCTAGGTGGCATCTGTATGCCCCTCTTATACGTTACCT 10000
## Hpl_1_0001 ..........................................A.......  1000
## Hpl_1_0002 .............................................C....   353
## Hpl_1_0003 .........................................C........   192
## Hpl_1_0004 ........C.........................................   125
## Hpl_1_0005 .....A............................................    89
## Hpl_1_0006 ..A...............................................    68
## Hpl_1_0007 ..........................A.......................    53
## Hpl_1_0008 ....................C.............................    37
## Hpl_1_0009 .....................................C............    27
## Hpl_1_0010 .........................................G........    19
## Hpl_1_0011 .........................................G........    17
## Hpl_1_0012 .................T................................    15
## Hpl_1_0013 .........................G........................    13
## Hpl_1_0014 ................................................A.    12
## Hpl_2_0001 ................A.......T.........................    44
## Hpl_2_0002 ...............G............................A.....    24
## Hpl_2_0003 ..............G..........C........................    21
## Hpl_2_0004 ..............C.........G.........................    14
## Hpl_2_0005 ......A.......G...................................    11
## Hpl_3_0001 ..............C.........A...............A.........   600
## Hpl_3_0002 ...................T....A...................A.....    31
## Hpl_3_0003 ........................A.............G.A.........    30
## Hpl_3_0004 ..............C.......................G.A.........    10
## Hpl_4_0001 ..............C.........A.............G.A.........  1000
## Hpl_4_0002 ..............A.........A.............G.A.........    24
## Hpl_5_0001 T.............C.........A.............G.A.........  3000
## Hpl_5_0002 ....C.........C.........A.............G.A.........   750
## Hpl_5_0003 ..............C.........A.............G.AT........   400
## Hpl_5_0004 ..............CT........A.............G.A.........   333
## Hpl_5_0005 ..............C.........AG............G.A.........   187
## Hpl_5_0006 ..............C.........A...........G.G.A.........   120
## Hpl_5_0007 ..............C.........A.............G.A.......T.    83
## Hpl_5_0008 ..............C.........A........T....G.A.........    61
## Hpl_5_0009 ..............C.........A..........G..G.A.........    46
## Hpl_5_0010 .G............C.........A.............G.A.........    37
## Hpl_5_0011 ..............C.........A..........A..G.A.........    20
## Hpl_5_0012 ..............C.......T.A.............G.A.........    17
## Hpl_5_0013 ..............C.........A.............G.A.....T...    15
## Hpl_5_0014 ..............C.........A.............G.A..T......    13
## Hpl_5_0015 ..............C.........A...........T.G.A.........    11
## Hpl_5_0016 ..............C.A.......A.............G.A.........     9
## Hpl_5_0017 ..............C.........A.............G.A.T.......     7
## Hpl_6_0001 ..............C.C..T....A.............G.A.........     8
```

The genome Hpl_4_0001 gives rise to the segregating population.

```r
cnm <- cumsum(table(lst$nm))+1
nm.pos <- as.vector(cnm)[-length(cnm)]
names(nm.pos) <- names(cnm[-1])
bp <- barplot(lst$nr,col=c("lavender","pink")[c(rep(1,22),rep(2,20))])
axis(1,at=bp[nm.pos],labels=names(nm.pos),cex.axis=0.7)
```

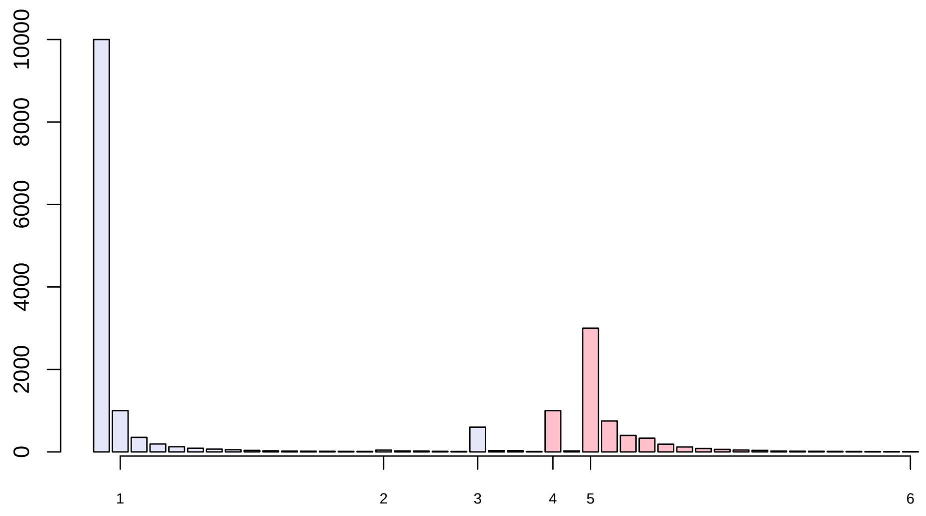


Figure 9: Simulated abundances with diverging populations

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=ca_ES.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=ca_ES.UTF-8        LC_COLLATE=ca_ES.UTF-8
##  [5] LC_MONETARY=ca_ES.UTF-8    LC_MESSAGES=ca_ES.UTF-8
##  [7] LC_PAPER=ca_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=ca_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
## [1] QSutils_0.99.0      Biostrings_2.46.0   XVector_0.18.0
## [4] IRanges_2.12.0      S4Vectors_0.16.0    BiocGenerics_0.24.0
## [7] devtools_1.13.6     BiocStyle_2.6.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17    knitr_1.20      magrittr_1.5    zlibbioc_1.24.0
##  [5] mnormt_1.5-5    ape_5.1         lattice_0.20-35 stringr_1.3.1
##  [9] tools_3.4.4     grid_3.4.4      nlme_3.1-137    xfun_0.3
## [13] psych_1.8.4     git2r_0.21.0    withr_2.1.2     htmltools_0.3.6
## [17] yaml_2.2.0      rprojroot_1.3-2 digest_0.6.15   bookdown_0.7.17
## [21] memoise_1.1.0   evaluate_0.10.1 rmarkdown_1.10  stringi_1.2.3
## [25] compiler_3.4.4  backports_1.1.2 foreign_0.8-70
```

References

Gregori, Josep, Celia Perales, Francisco Rodriguez-Frias, Juan I. Esteban, Josep Quer, and Esteban Domingo. 2016. "Viral quasispecies complexity measures." *Virology* 493. Elsevier: 227–37. doi:10.1016/j.virol.2016.03.017.

Gregori, Josep, Miquel Salicrú, Esteban Domingo, Alex Sanchez, Juan I. Esteban, Francisco Rodríguez-Frías, and Josep Quer. 2014. "Inference with viral quasispecies diversity indices: Clonal and NGS approaches." *Bioinformatics* 30 (8): 1104–11. doi:10.1093/bioinformatics/btt768.